Sylva : plate-forme de validation multi-niveaux de lexiques

Karen Fort, Bruno Guillaume

TALN 2008, Avignon, 9–13 juin 2008

Sylva : plate-forme de validation multi-niveaux de lexiques

Karën Fort Bruno Guillaume
LORIA / INRIA Nancy Grand-Est
Karen.Fort@loria.fr, Bruno.Guillaume@loria.fr

Résumé. La production de lexiques est une activité indispensable mais complexe, qui nécessite, quelle que soit la méthode de création utilisée (acquisition automatique ou manuelle), une validation humaine. Nous proposons dans ce but une plate-forme Web librement disponible, appelée Sylva (Systematic lexicon validator). Cette plate-forme a pour caractéristiques principales de permettre une validation multi-niveaux (par des validateurs, puis un expert) et une traçabilité de la ressource. La tâche de l'expert(e) linguiste en est allégée puisqu'il ne lui reste à considérer que les données sur lesquelles il n'y a pas d'accord inter-validateurs.

Abstract. Lexicon production is essential but complex and all creation methods (automatic acquisition or manual creation) require human validation. For this purpose, we propose a freely available Web-based framework, named Sylva (Systematic lexicon validator). The main point of our framework is that it handles multi-level validations and keeps track of the resource's history. The expert linguist task is made easier : (s)he has only to consider data on which validators disagree.

Mots-clés : Lexiques, plate-forme de validation, cadres de sous-catégorisation.

Keywords: Lexicons, validation framework, subcategorization frames.

1 Introduction

De nombreux outils de traitement automatique des langues (TAL) nécessitent des ressources linguistiques et notamment des lexiques. La construction de ce type de ressources présente de nombreuses difficultés :
– différents types de lexiques sont nécessaires en fonction des applications ;
– le nombre d'entrées à décrire est souvent important (de l'ordre de 500 000 pour les formes fléchies du français par exemple) ;
– suivant la tâche visée, des informations plus ou moins détaillées sont nécessaires ;
– la façon de coder le contenu linguistique doit parfois être adaptée aux outils ou au moins aux théories mises en jeu dans ces outils.

Pour la langue française, les lexiques morphologiques sont moins problématiques : ce type de lexique ne pose pas de problème majeur quant au contenu linguistique qu'il doit contenir. On dispose ainsi de lexiques morphologiques de bonne qualité avec une couverture satisfaisante (Morphalou (Romary *et al.*, 2004), partie morphologie du Le*fff* (Sagot *et al.*, 2006), ...). Le problème est plus aigu pour les lexiques syntaxiques sur lesquels nous nous concentrons ici. Des acteurs du TAL francophones travaillent ensemble régulièrement depuis deux ans sur le développement de ressources syntaxiques du français (projet LexSynt[1]), le travail présenté ici pour la validation est en partie issu des reflexions menées dans le cadre de ce projet. Il est important de noter que la plate-forme présentée n'est pas spécifique à ces lexiques et pourrait être utilisée avec des lexiques sémantiques pour lesquels les problèmes sont encore plus marqués.

La construction manuelle de ressources par des experts linguistes est a priori la plus satisfaisante, mais la quantité de travail nécessaire rend cette méthode irréaliste dans la plupart des cas. Quelques exemples de ressources manuelles existent néammoins : pour le français, on peut citer DICOVALENCE (van den Eynde & Mertens, 2003), les tables du lexique-grammaire (Gross, 1975) ou la base lexicale de Dubois (Dubois & Dubois-Charlier, 1997). À l'opposé, des méthodes de construction automatique de lexiques à partir de corpus ont été proposées : acquisition de cadres de sous-catégorisation à partir de corpus (Preiss *et al.*, 2007) ou de corpus arborés (Kupsc, 2007). L'inconvénient de cette dernière méthode est évidemment la qualité du lexique produit. En effet, la pertinence de chacune des entrées ne peut être qu'estimée et la couverture dépend de la taille des corpus utilisés et de leur représentativité pour l'aspect considéré dans le lexique.

Dans la pratique, le plus souvent, la production de lexique utilise des méthodes mixtes qui automatisent une partie du travail et qui minimisent l'intervention humaine. L'acquisition automatique se fait sur des ressources existantes : des corpus bruts ou arborés, des dictionnaires électroniques (construction de Morphalou à partir du TLFi (Pierrel *et al.*, 2004)), ou d'autres ressources informatisées mais qu'il faut adapter (contruction de SynLex (Gardent *et al.*, 2006) à partir des tables du lexique-grammaire). La ressource produite dépend alors de la qualité de la ressource de départ et de la méthode de conversion. Dans le cas de SynLex, la ressource produite souffre de nombreuses imperfections (Falk *et al.*, 2007) et l'intervention humaine est nécessaire pour valider a posteriori toute ou partie de la ressource.

La validation multi-niveaux permet de reporter une partie du travail humain vers des linguistes moins expérimentés. Le premier niveau de validation est ainsi réalisé par plusieurs validateurs non-experts, le deuxième niveau permet ensuite au linguiste expert de ne considérer que les cas où les différents validateurs ont des avis divergents.

[1] `http://lexsynt.inria.fr`

Nous proposons ici une plate-forme pour gérer la validation multi-niveaux de lexiques. Cette plate-forme permet la gestion des deux types d'utilisateurs et automatise le passage par les deux niveaux de validation. Après avoir décrit les fonctionnalités et l'implantation de la plate-forme, nous montrons comment elle peut s'instancier avec un lexique syntaxique (SynLex) produit automatiquement à partir des tables du lexique-grammaire.

2 Choix méthodologiques

Notre objectif premier est de permettre la validation rapide et complète d'une ressource à large couverture. C'est cet objectif qui a guidé certains choix pour cette première version de l'outil.

2.1 Interface simplifiée

Pour chaque entrée du lexique, le validateur a le choix entre trois jugements : *accepté*, *refusé*, *ne sait pas*. Il n'a donc pas la possibilité d'éditer une entrée pour la modifier. En effet, le but de cette interface est de permettre au validateur de se prononcer rapidement sur chacune des entrées. Nous sommes bien conscients que cela n'est pas toujours satisfaisant et que certaines entrées devraient être corrigées plutôt que rejetées. C'est pourquoi, il est demandé au validateur de motiver ses choix par des commentaires guidés. Il sera ainsi possible d'envisager une phase ultérieure de correction des entrées qui posent problème, guidée par les commentaires de la première phase.

Une autre conséquence de cette interface simplifiée sans possibilité d'édition est que chaque entrée doit contenir une information suffisamment atomique. Par exemple, il n'est pas possible de présenter des valences (ou cadres de sous-catégorisation, CSC) différentes avec une notation factorisée en une seule entrée.

2.2 Ajout d'entrées a posteriori

Il n'est pas prévu pour l'instant d'ajouter des informations dans la ressource. En effet, la notion de paquet (décrite en section 4.2), qui permet d'obtenir une série de vues transversales sur la ressource, interdit le regroupement des entrées par lemme. Par conséquent, le validateur ne peut se faire une idée sur l'ensemble des cadres d'un lemme et il ne peut donc pas juger de la couverture du lexique pour un lemme donné. Ainsi, si l'ajout d'entrées existait, il ne pourrait pas être systématique et ne serait pas homogène.

Le problème de complétion par des entrées manquantes nécessite selon nous d'autres méthodes comme l'intégration d'autres ressources (Dicovalence, Le*fff*, ...) ou l'étude sur corpus (éventuellement guidée par un ou plusieurs analyseurs syntaxiques).

2.3 Lien facultatif vers des exemples

Dans la version actuelle de l'outil, l'ajout d'un exemple est facultatif lorsque l'on accepte un cadre. Des expériences sont prévues pour comparer la vitesse de validation avec ou sans ajout d'exemple.

Quoi qu'il en soit, nous pensons qu'une validation, même sans exemple, produira une ressource avec une forte valeur ajoutée et qu'il est possible dans une deuxième phase de combiner l'amélioration de la couverture et l'ajout d'exemples.

Dans cette seconde phase, il sera sans doute plus efficace de proposer des exemples et de demander au validateur de choisir le CSC qui correspond (éventuellement guidé par un ou plusieurs analyseur(s) syntaxique(s)), plutôt que de lui demander de fournir lui-même un exemple.

3 La plate-forme Sylva

3.1 Fonctionnalités

La plate-forme Sylva est bâtie autour d'une base de données, conçue pour gérer à la fois les informations linguistiques du lexique et les informations périphériques (source, historique de validation, gestion des paquets (décrits en section 4.2)).

Nous donnons ci-dessous, par ordre d'importance décroissant, les principales fonctionnalités de Sylva[2] :

Validation multi-niveaux. Comme on l'a vu, il est difficile de demander à des experts linguistes de valider un lexique dans son intégralité. Nous proposons donc d'utiliser comme validateurs de premier niveau des linguistes non-experts (des étudiant(e)s en Master Sciences du Langage, par exemple). Ainsi, dans la version actuelle de Sylva, chaque entrée est validée par deux validateurs différents. Lorsque les deux rendent le même verdict (positif ou négatif), le jugement est automatiquement appliqué à l'entrée. Lorsque les jugements diffèrent ou lorsque les validateurs hésitent, Sylva demande à l'expert linguiste de trancher[3].

Traçabilité. Nous avons également décidé de conserver le plus d'information possible pour chaque entrée. Outre la description linguistique, nous conservons dans la base de données les informations concernant la source de la ressource initiale (d'où proviennent les données utilisées pour peupler la base) ainsi que celles concernant les processus de validation (le nom des validateurs, leur jugement, le jugement de l'expert s'il y a lieu, ...).

Vue transversale. Le nombre d'entrées à valider peut être relativement important : le premier lexique que nous allons valider avec Sylva contient par exemple près de 30 000 entrées. Une validation globale de la ressource serait problématique : par exemple, il serait difficile pour l'expert de répartir les tâches aux différents validateurs. Il est donc possible de décomposer la ressource en ensembles plus petits, linguistiquement cohérents. Dans Sylva, ces ensembles correspondent à la notion de *paquet* (voir section 4.2).

3.2 Acteurs

L'interface utilisateur est accessible sur le Web. Trois types d'acteurs existent : l'*invité*, le *validateur* et l'*expert*. Pour ces deux derniers, la connexion se fait par identifiant et mot de passe.

[2] la version en cours de développement est disponible ici : `http://sylva.loria.fr/`

[3] Sylva est prévu pour permettre la validation par plus de deux validateurs, il faut dans ce cas paramétrer les règles de décisions pour l'accord (unanimité, majorité, ...)

3.2.1 L'utilisateur invité

L'invité peut :
– voir la ressource dans son état actuel. Plusieurs types de vues sont disponibles, par lemme ou par initiale, par exemple. Il sera également possible à terme de filtrer les entrées selon des critères plus fins (verbes possédant tel type de valence, verbes dont le lemme correspond à une expression rationnelle donnée, etc.).
– exporter le lexique dans un format texte ou XML respectant la norme Lexical Markup Framework (Francopoulo *et al.*, 2006).
– poster des questions, des suggestions via une interface d'envoi de courriers électroniques.

3.2.2 Le validateur

Le rôle du validateur est de juger toutes les entrées des paquets qui lui sont assignés. Pour chaque paquet, une page Web s'affiche. Les informations concernant une entrée sont présentées de manière condensée sur une seule ligne. Le validateur peut :
– accepter l'entrée et éventuellement en donner un exemple d'utilisation,
– rejeter l'entrée en donnant une explication pour ce rejet (l'interface de validation suggère certaines explications afin d'aider le validateur à formuler plus rapidement son jugement),
– exprimer un doute (*Ne sait pas*).
En outre, quelle que soit sa décision, le validateur peut toujours ajouter un commentaire.

3.2.3 L'expert

Les principaux rôles de l'expert sont de :
– créer et gérer les comptes des validateurs, et ce afin d'assurer la souplesse de validation,
– assigner les paquets aux validateurs,
– finaliser la validation d'un paquet une fois les validations de premier niveau terminées, c'est-à-dire trancher les cas pour lesquels il n'y a pas d'accord inter-validateurs.
Il est important de noter que l'expert peut également valider directement les entrées, court-circuitant ainsi les validateurs. Cette fonctionnalité a été prévue pour faire face à des cas exceptionnels, comme par exemple l'incapacité pour un validateur de terminer sa tâche ou pour accélérer la validation.

3.3 Qualité

Sylva a été conçue et développée dans le respect des techniques de développement les plus récentes. En particulier, cette plate-forme a fait l'objet de spécifications UML (Unified Modeling Language) détaillées, validées par les utilisateurs. Nous avons par ailleurs utilisé un *framework* de développement PHP robuste, bien documenté et très utilisé, *symfony* (Potencier & Zaninotto, 2007), qui offre :

– séparation modèle/objet et Modèle/Vue/Contrôleur (Gamma *et al.*, 1995),
– indépendance à l'implémentation de la base de données (Object Relational Mapping),
– le support du multilinguisme (encodage UTF-8) et de l'internationalisation,
– une couche sécurité performante, permettant de protéger la base contre les intrusions,

– des performances accrues grâce à l'utilisation d'AJAX (Asynchronous JavaScript And XML).

Cela facilite une conception propre et l'écriture d'un code lisible, rendant de ce fait la maintenance, les modifications et la localisation plus faciles.

Par ailleurs, une documentation utilisateur détaillée et adaptée au statut de l'utilisateur est disponible en ligne, afin de permettre à la fois à l'expert linguiste et aux validateurs de bénéficier au mieux des fonctionnalités de Sylva. Une documentation développeur sera également fournie pour permettre d'adapter l'outil à d'autres types de ressources.

4 Un exemple de mise en œuvre de Sylva

Une instanciation de la plate-forme est une description précise des informations associées aux entrées du lexique. À partir de cette description, nous devons adapter l'interface utilisateur correspondante pour la validation. Il faut notamment adapter la partie de l'interface qui permet au validateur de motiver son jugement en cas de refus.

4.1 La ressource utilisée

La validation à grande échelle d'une première ressource est en cours. Nous avons choisi de commencer par le lexique syntaxique SynLex (Gardent *et al.*, 2006). En effet, malgré des défauts identifiés (Falk *et al.*, 2007), SynLex est construit à partir d'une ressource de qualité (les tables du lexique-grammaire français (Gross, 1975)) qui est l'une des seules à concilier couverture large et informations détaillées. Il est à noter que SynLex ne fait pas de distinction de sens pour un même lemme car il a été construit uniquement par extraction des informations syntaxiques contenues dans les tables du lexique-grammaire.

La validation de SynLex va permettre d'en améliorer la précision ; l'amélioration de la couverture se fera à l'aide d'autres méthodes (cf. section suivante).

Dans Synlex, une entrée se compose d'un verbe, d'une liste d'arguments syntaxiques ayant un rôle sémantique (*a0*, *a1*, ...), d'une liste optionnelle d'associés (arguments régis par le verbe mais ne remplissant pas de rôle sémantique, *Ilimp* pour « il impersonnel », par exemple) et d'une liste de macros (informations supplémentaires sur les propriétés syntaxiques du verbe comme l'auxiliaire ou la passivisation). Les associés et les macros sont des listes finies d'atomes. Chaque argument est décrit par : fonction grammaticale (*suj*, *objde*, ...), marqueur optionnel (*de*, *pour*, ...), catégorie syntaxique (*scompl*, *sn*, ...), restrictions morphologiques (pluriel obligatoire, par exemple), restrictions sémantiques (*humain*, *non humain*, *abstrait*).

Construit à partir de 60% des tables, la version de SynLex utilisée pour peupler la base décrit 3 494 verbes différents et 26 714 entrées. La figure ci-dessous décrit quelques entrées associées au lemme « abuser » (celles qui apparaissent dans l'interface de la figure 2), ainsi que quelques entrées associées au lemme « épater » qui font apparaître d'autres éléments (marqueur *de*, macro *être*).

```
abuser <a0:suj:sinf:::>
abuser <a0:suj:sn:::humain>
abuser <a0:suj:sn:::non_humain>
abuser <a0:suj:scompl:::>
abuser <a0:suj:sn:::non_humain,a1:obj:sn:::humain>
abuser <a0:suj:sinf:::,a1:obj:sn:::humain>
abuser <a0:suj:sn:::humain,a1:obj:sn:::humain>
abuser <a0:suj:scompl:::,a1:obj:sn:::humain>

épater <a1:suj:sn:::humain,a0:objde:de-scompl:::>; etre, participe-passe
épater <a1:suj:sn:::humain,a0:objde:de-sn:::abstrait>; etre, participe-passe
```

FIG. 1 – Exemples d'entrées de SynLex.

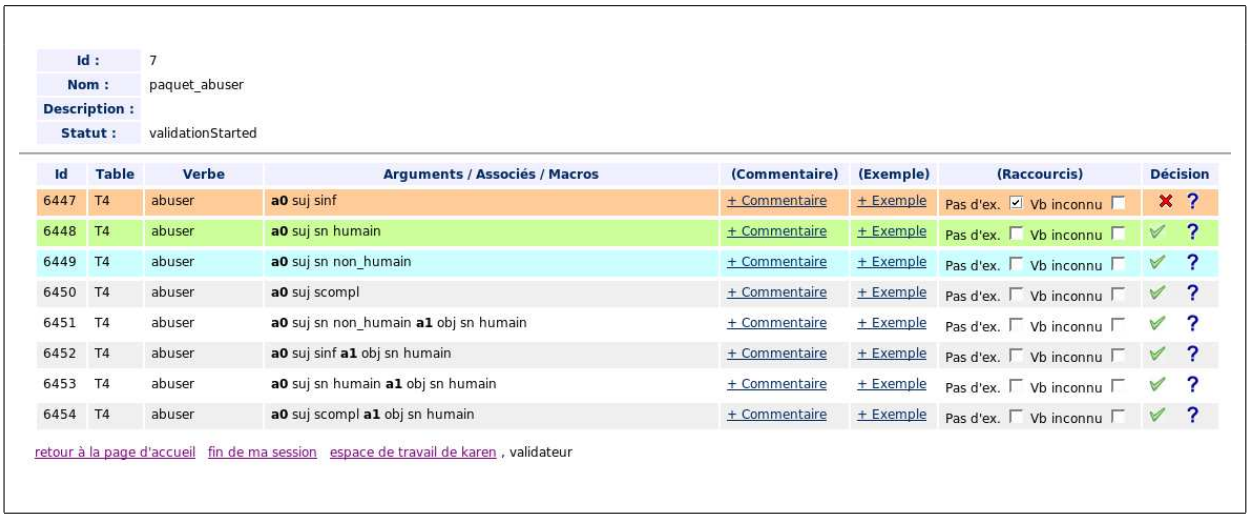


FIG. 2 – Interface du validateur pour SynLex (version béta).

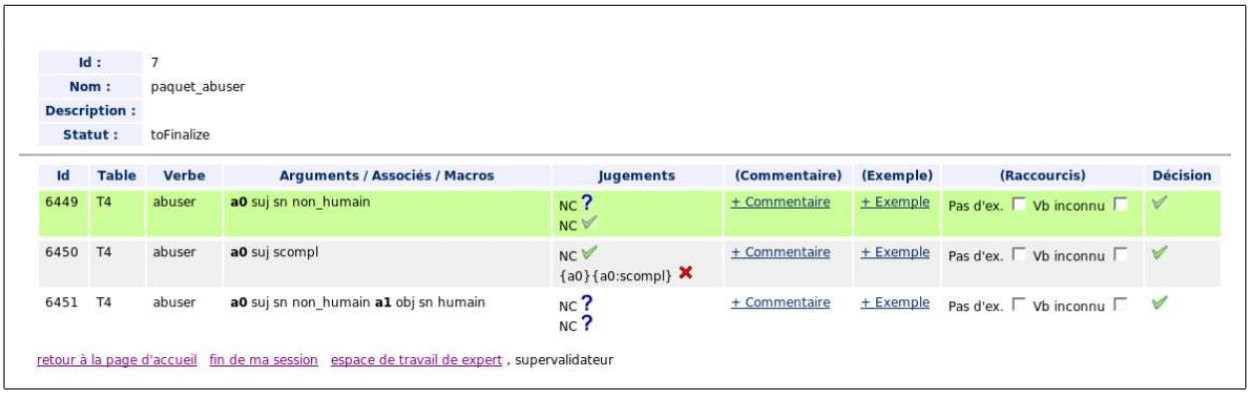
4.2 Les paquets

Afin de rendre la phase de validation la plus efficace possible, la ressource est décomposée en ensembles, linguistiquement cohérents et de taille raisonnable (entre 50 et 210 entrées), les paquets. Etant donnée la taille de SynLex, ceux-ci ont été construits de manière automatique, avant le peuplement de la base, notre but étant de réunir les entrées ayant un comportement semblable, afin d'en simplifier la validation.

Une première étape de tri consiste à regrouper les entrées possédant le même CSC. Les groupes suffisamment gros forment directement des paquets : 109 paquets sont produits ainsi. Les entrées restantes sont regroupées par table d'origine : 110 paquets. Enfin, le reste est arbitrairement réparti dans les 9 derniers paquets. Au final, 228 paquets peuplent la base.

4.3 Des interfaces adaptées

La validation de SynLex a donné lieu au développement d'interfaces spécifiques, qui, grâce à la séparation Modèle/Vue/Contrôleur sont facilement adaptables à d'autres lexiques. Des captures d'écrans des interfaces de validation des validateurs et de l'expert pour la ressource actuelle sont présentées respectivement en figures 2 et 3. L'interface de gestion de l'expert est quant à elle présentée en figure 4.



Id :	7
Nom :	paquet_abuser
Description :	
Statut :	toFinalize

Id	Table	Verbe	Arguments / Associés / Macros	Jugements	(Commentaire)	(Exemple)	(Raccourcis)	Décision
6449	T4	abuser	**a0** suj sn non_humain	NC ? NC ✓	+ Commentaire	+ Exemple	Pas d'ex. ☐ Vb inconnu ☐	✓
6450	T4	abuser	**a0** suj scompl	NC ✓ {a0}{a0:scompl} ✗	+ Commentaire	+ Exemple	Pas d'ex. ☐ Vb inconnu ☐	✓
6451	T4	abuser	**a0** suj sn non_humain **a1** obj sn humain	NC ? NC ?	+ Commentaire	+ Exemple	Pas d'ex. ☐ Vb inconnu ☐	✓

retour à la page d'accueil fin de ma session espace de travail de expert , supervalidateur

FIG. 3 – Interface de l'expert pour SynLex (version béta).

supervalidateur: **expert**

Statistiques générales :

Paquets créés	**7**
Paquets assignés	**1**
Paquets en cours de validation	**4**
Paquets validés	**1**
Validateurs	**2**

Statistiques personnelles :

Paquets tranchés	**0**
Paquets en cours	**0**
Paquets à trancher	**1**

Paquets à trancher/en cours :

7 paquet_abuser toFinalize (62.5% d'accord)

Activités de gestion :

Validateurs :
Créer validateur
Voir/Modifier validateurs

Paquets :
Créer paquet
Voir/Modifier paquets
Assigner paquets à validateurs

retour à la page d'accueil fin de ma session espace de travail de expert , supervalidateur

FIG. 4 – Interface générale de l'expert pour SynLex (version béta).

5 Développements futurs

5.1 Intégration d'autres ressources

SynLex a été contruit à partir d'une ressource de grande qualité, produite manuellement, mais il reste limité à une partie des verbes (seules 60% des tables de lexique-grammaire des verbes sont disponibles librement). Il est par conséquent nécessaire de le fusionner avec d'autres ressources pour obtenir un lexique syntaxique du français générique.

Le Le*fff* (lexique des formes fléchies du français) est un lexique du français à large couverture qui fournit des informations morphologiques et syntaxiques, non seulement pour les verbes, mais aussi pour toutes les autres catégories morphosyntaxiques. Le Le*fff* surgénère souvent : par défaut, à tout nom sont affectés tous les CSC possibles pour les noms. Cette ressource est donc une bonne candidate pour une validation par filtrage des entrées. Le format du Le*fff* étant très proche de celui de SynLex, il peut être entré facilement dans la plate-forme actuelle. Cependant, les informations de sous-catégorisation sont très factorisées dans le Le*fff*, il faudra en tenir compte lors de son intégration soit en dépliant les cadres factorisés, soit en permettant au validateur de décrire plus finement les motivations de son choix.

Bien que nécessitant un travail linguistique conséquent, la traduction des informations de DICOVALENCE (van den Eynde & Mertens, 2003), lexique de grande qualité des valences verbales du français, permettrait d'améliorer significativement la qualité du lexique envisagé.

5.2 Intégration d'autres fonctionnalités

Comme nous l'avons déjà évoqué, la première fonctionnalité que nous souhaitons ajouter à Sylva est une interface Web permettant de relier des usages de verbes à des exemples. Encore une fois, il s'agit là d'une tâche longue et pénible. Nous envisageons d'utiliser pour cela des exemples extraits de corpus arborés, de corpus de textes ou de dictionnaires. Pour chaque exemple, l'utilisateur devra sélectionner l'une des entrées du lexique. Grâce aux informations fournies par le corpus arboré ou par le résultat d'une analyse syntaxique, on pourra réduire le nombre de possibilités proposées au validateur et ainsi lui faciliter la tâche.

Une autre fonctionnalité importante serait la possibilité de modifier les entrées. En utilisant les commentaires accompagnant les refus de la première phase, on pourra proposer au validateur des hypothèses de corrections de l'entrée parmi lesquelles il pourra choisir.

On peut imaginer que des entrées manquantes pourraient être détectées, y compris après l'intégration d'autres ressources, par exemple dans la phase de liage des exemples à partir de corpus. Il faudrait donc prévoir une interface pour ajouter de nouvelles entrées, au moins pour l'utilisateur expert.

Le développement de ces fonctionnalités nécessite un travail non négligeable, mais nous pensons que notre plate-forme est suffisamment générique et adaptable pour en faciliter et en accélérer l'implémentation.

Remerciements

Ce travail a été réalisé grâce à des fonds provenant du CPER MISN de la Région Lorraine. Les auteurs souhaitent remercier les autres membres du projet pour leur participation active aux spécifications : Claire Gardent, Guy Perrier et Ingrid Falk. Nous remercions également Mathieu Morey et François-Régis Chaumartin pour leur aide lors de la phase de développement.

Références

DUBOIS J. & DUBOIS-CHARLIER F. (1997). *Les Verbes Français*. Paris, France : Larousse-Bordas.

FALK I., FRANCOPOULO G. & GARDENT C. (2007). Évaluer SynLex. In *Actes de TALN 07*, p. 335–344, Toulouse.

FRANCOPOULO G., GEORGE M., CALZOLARI N., MONACHINI M., BEL N., PET M. & SORIA C. (2006). Lexical Markup Framework (LMF). In *Actes de LREC 06*, Gène, Italie.

GAMMA E., HELM R., JOHNSON R. & VLISSIDES J. (1995). *Design Patterns : Elements of Reusable Object-Oriented Software*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc.

GARDENT C., GUILLAUME B., PERRIER G. & FALK I. (2006). Extraction d'information de sous-catégorisation à partir des tables du LADL. In *Actes de TALN 06*, p. 139–148, Louvain.

GROSS M. (1975). *Méthodes en syntaxe*. Hermann.

KUPSC A. (2007). Extraction automatique de cadres de sous-catégorisation verbale pour le français à partir d'un corpus arboré. In *Actes de TALN 07*, Toulouse, France.

PIERREL J., DENDIEN J. & BERNARD P. (2004). Le TLFi ou Trésor de la Langue Française informatisé. In *Actes de EURALEX 04*, Lorient, France.

POTENCIER F. & ZANINOTTO F. (2007). *The Definitive Guide to symfony*. Berkeley, CA, USA : Apress.

PREISS J., BRISCOE T. & KORHONEN A. (2007). A system for large-scale acquisition of verbal, nominal and adjectival subcategorization frames from corpora. In *Actes de ACL 07*, Prague, République tchèque.

ROMARY L., SALMON-ALT S. & FRANCOPOULO G. (2004). Standards going concrete : from LMF to Morphalou. In M. ZOCK, Ed., *COLING 2004 Enhancing and using electronic dictionaries*, p. 22–28, Genève, Suisse.

SAGOT B., CLÉMENT L., VILLEMONTE DE LA CLERGERIE É. & BOULLIER P. (2006). The Lefff 2 syntactic lexicon for French : architecture, acquisition, use. In *Actes de LREC 06*, Gène, Italie.

VAN DEN EYNDE K. & MERTENS P. (2003). La valence : l'approche pronominale et son application au lexique verbal. *Journal of French Language Studies*, **13**, 63–104.